EE 374: Fundamentals of Blockchain Infrastructure **Stanford, Spring 2025**

Lecture 5: Bitcoin Safety and Liveness

April 14, 2025

Lecturer: Prof. David Tse Scribe: Abhiram Gorle

1 Introduction

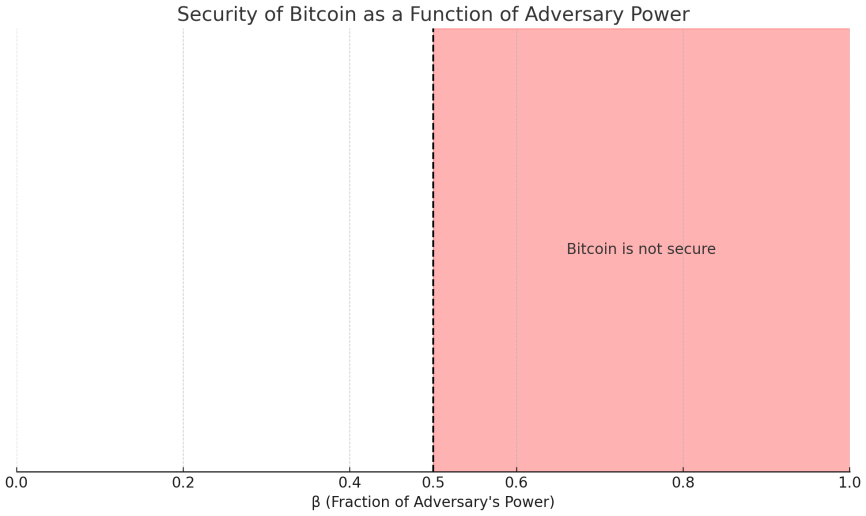
This lecture delves into the foundational properties of **safety** and **liveness** in the context of the Bitcoin protocol. It begins by formally defining these properties within distributed consensus and explores the specific conditions under which Bitcoin upholds them. The discussion emphasizes how these guarantees are maintained even in adversarial environments, by first analysing a specific attack, the private attack, and then arguing that attack is the worst attack. Through this lens, the security of the Bitcoin protocol is analyzed.

Property	**Intuition**
Safety	Nothing bad ever happens. Once a transaction is considered confirmed by a client, all clients will always consider it confirmed (prevents de-confirmation).
Liveness	Something good eventually happens. Every valid transaction broadcast by an honest party will, after finite time, appear permanently in the ledger.

Table 1: Core properties of consensus protocols.

2 Recap

In Lecture 3, we examined the cryptographic foundations and security guarantees of the proof-of-work (PoW) longest chain protocol, with a focus on potential attacks and their analysis. We reviewed how hash functions—specifically SHA-256—are used to construct blocks and secure consensus, relying on their collision resistance and the stronger random oracle model to ensure unpredictability and difficulty in mining. The core threat model explored was the **private double-spend attack**, in which an adversary secretly mines a longer chain to reverse a previously confirmed transaction. To mitigate this, Bitcoin implements a k**-deep confirmation rule**, only treating transactions as finalized once buried under k blocks in the longest chain. We analyzed the probability of such attacks using the *Nakamoto Race*, modeling mining as a Poisson process and showing that the likelihood of a successful attack drops exponentially with increasing k, provided honest miners control the majority of hash power. We assume the network delay to be negligible throughout this analysis.



3 Notations and Definitions

We first define the key quantities and notation used throughout the lecture.

- λ_h — the effective mining rate of *honest miners*, i.e., the rate at which honest nodes collectively find blocks, λ_a — the effective mining rate of the *adversary*, i.e., the rate at which adversarial miners generate blocks.
- β — The fraction of total mining power controlled by the adversary. That is,

$$\beta = \frac{\lambda_a}{\lambda_a + \lambda_h} \quad \text{and} \quad 1 - \beta = \frac{\lambda_h}{\lambda_a + \lambda_h}$$

  Now $0 \leq \beta \leq 1$, and for security, that $\beta < \frac{1}{2}$. (if $\beta \geq \frac{1}{2}$, private attack is successful with a non-vanishing probability)

- **Clients** — Participants in the blockchain system who submit and confirm transactions. While *miners* (nodes that contribute hash power) are a special class of clients, *not all clients are miners*. Many users interact with the system without participating in block production.
- **Chain Growth**. *For Bitcoin, we define the chain growth of a longest chain $\mathcal{C}$ as the average growth rate of $\mathcal{C}$ (number of blocks per unit time), denoted as $CG(\mathcal{C})$.*
- **Chain Quality**. *For Bitcoin, we define the chain quality of a longest chain $\mathcal{C}$ as the fraction of honest blocks in $\mathcal{C}$, denoted as $CQ(\mathcal{C})$.*

4 Safety of Bitcoin

In the Nakamoto consensus protocol model, we assume that there is a single adversary and many different honest parties participating in the protocol with the adversary's computing power being a fraction β of the total computing power of all miners in the system. Recall the k-deep confirmation

rule in Bitcoin: a node treats all but the last $k-1$ blocks in its longest chain as confirmed. Once we confirm a block, we also confirm all the transactions in it. Note that the confirmation rule is *locally* applied by each client; therefore, a transaction confirmed by one client needn't be confirmed by another. However, it is desirable that a transaction confirmed by one client is soon confirmed by all other clients, and remains confirmed forever after (without any *de-confirmation*). In blockchains, this is what we call the *safety* property. To get a sense of how the adversary can disrupt safety, let us take a look at some possible adversarial actions in the adversary's attack space.

4.1 Private Attack

This is the attack considered by Nakamoto. In summary, the adversary mines on a *private fork* that is hidden from honest miners until it becomes at least as long as the public chain; upon release, it overwrites honest history (double-spend).

Suppose the adversary wishes to violate the safety of a block. To do so, the adversary must ensure that block is first confirmed by some (or all) clients, and then, at some time in the future, must dislodge B from the longest chain, i.e., it must create a fork from a block preceding B, and must eventually produce a chain of length equal to or longer than the longest chain containing B, after B has been confirmed.

When should the adversary reveal its chain to the honest clients? Suppose it reveals its private chain while it is shorter than the longest honest chain. The honest clients will simply ignore the adversary's chain, and it will not produce any effect. Thus, the adversary must reveal its chain only when it is at least as long as the honest chain. What happens if the adversary reveals its chain too early, i.e., before block gets k-deep? It would still end up displacing, but then its actions do not lead to a safety violation! Thus, the adversary must also wait until the honest chain is long enough. (see Figure 1)

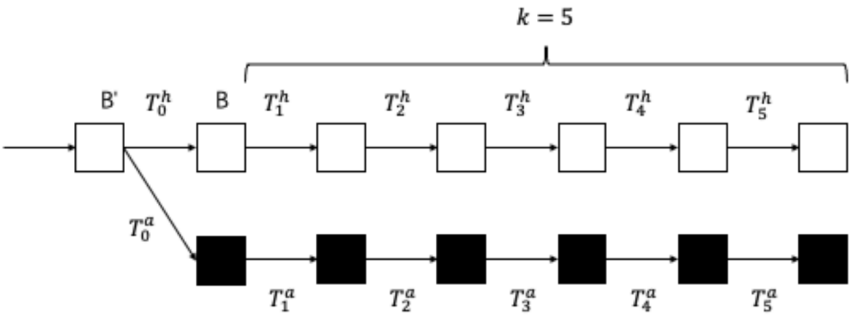


Figure 1: Private attack on block B with $k = 5$

4.2 Balance Attack

Another possible attack from the large space of adversarial actions is the *balance attack*, illustrated in Figure 2. The adversary strategically mines and publishes blocks to maintain *two* competing forks of nearly equal length, continuously shifting honest miners' hash power between them. Unlike the previous case, the adversary tries to interact with the honest miners in this case.

(Figures 2, 1 taken from [3].)

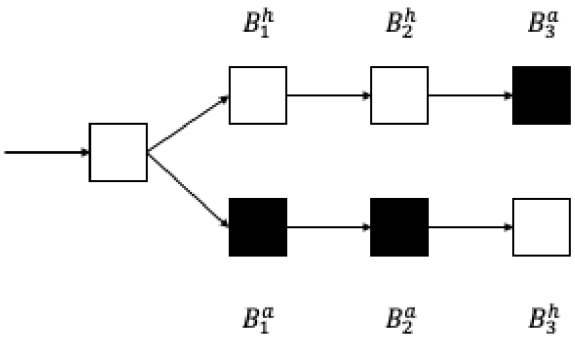


Figure 2: Illustrating a Balance Attack

4.3 Private attack is the worst-case attack

An attacker might first launch a balance attack to split honest mining power between two chains, slowing their growth, and then follow up with a fatal private attack. In general, adversarial strategies vary widely, defined by two key choices: *where* to mine and *when* to publish. However, in our case of zero network delay, we will show that any successful safety attack **implies** a successful private attack, along any sample paths of mining times of the underlying Poisson processes. This analysis is based on some key observations that we illustrate using Figure 3.

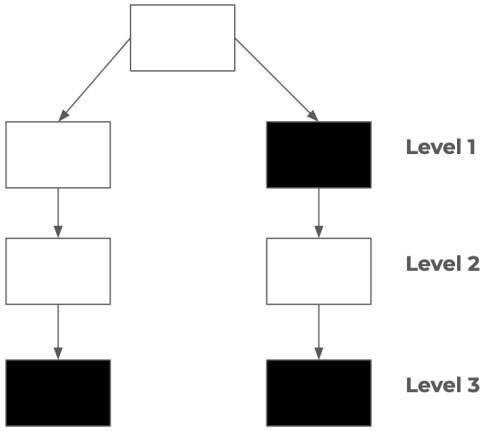


Figure 3: A fictitious two-fork network

The white blocks are mined by honest miners, and the black blocks by adversary miners. Note that such a configuration is not possible because at Level 2, once the first white block is mined, it is made public and therefore it becomes the tip of the longest chain, and all honest miners should mine on level 3 instead of continuing to mine on level 2, which would be the only way by which the second white block at level 2 is mined.

In this analysis, we only focus on a deconfirmation event at level 1. Now, suppose at time $t = T$, de-confirmation happens. Consider two random variables A, H which represent the number of blocks mined by the adversary and the honest nodes respectively up til time T. Then, we have the

following conditions:

- **Condition 1:** The total number of blocks mined is $A+H$ must be at least $2k$ since to launch a safety attack at least two chains of length $k+1$ must be present: one of the chains of length $k+1$ must have been present to confirm the block B and another chain of length $k+1$ must be present to deconfirm the block B by switching the longest chain. Ignoring the genesis block, $A+H \geq 2k$.
- **Condition 2:** We note that **two honest nodes can never be mined** (refer Figure 3) at the same level of the blockchain; this is because the network delay is zero and miners instantaneously learn of any newly mined block and all honest nodes have full consensus on the longest chain. Since honest nodes mine on the tip of the longest chain, there could not be parallel mining by the honest nodes at the same level. Hence, $A \geq H$.

Combining these two conditions implies also $A \geq k$ Now consider the above block-mining times sample path and imagine the adversary had *withheld* every one of its blocks until T. If $H \geq k$, then the adversary an release all Av blocks, and a deconfirmation event happens at this time T. If $H < k$, then the attack waits until the honest chain is equal to k, then release the A blocks, again successfully deconfirming the tx confirmed on level 1. So private attack would have succeeded.

This implies that if any attack succeeds, then a private attack would have succeeded. So the deconfimation error probability of any attack must be upperbounded by that of a private attack. In that sense, the private attack is the worst attack.

Catch-up probability for private forks: Suppose the adversary lags the public chain by k blocks. Classic gambler's ruin or random walk arguments give the probability that it *ever* catches up: (as described in [2])

$$P_{\text{reorg}}(k,\beta) = \begin{cases} \left(\frac{\beta}{1-\beta}\right)^k, & \beta \leq \frac{1}{2}, \\ 1, & \beta \geq \frac{1}{2}. \end{cases}$$

Hence for any fixed $\boldsymbol{q} < \frac{1}{2}$, the failure probability decays exponentially in k.

Formally, let $q_{k,\pi}$ denote the probability of k-deep confirmation under an attack π, then:

$$q_{k,\pi} \leq \left(\frac{\beta}{1-\beta}\right)^k = q_{k,\textit{private attack}},$$

the classic private-attack probability described above. And choosing k so that the right-hand side is negligible secures Bitcoin against *all* possible adversarial attacks.

5 Liveness of Bitcoin

While safety is an important security property of a blockchain protocol, an equally important one is *liveness*: this is the event that (honest) transactions get included into blocks, and further that the blocks feature in the longest chain. Liveness ensures that all transactions make their way into the ledger and safety ensures that eventually the transactions stay permanently in the ledger (with high probability). Together, liveness and safety ensure the security of the blockchain protocol.

We have already defined **chain growth** (CG) and **chain quality** (CQ) in Section 3. We also note that chain quality is typically defined under worst-case adversarial conditions, and usually the larger the chain quality, the better.

5.1 Liveness via Chain Quality

Let $\mathcal{C}$ be the current longest chain maintained by honest nodes. A blockchain protocol satisfies liveness if there exists a constant $\mu > 0$ such that for any sufficiently long segment of $\mathcal{C}$, the fraction of honestly mined blocks is at least μ. That is, $\text{CQ}(\mathcal{C}) \geq \mu > 0$. [1]

Recall that the *chain quality* of the longest chain $\mathcal{C}$ is defined as

$$\text{CQ}(\mathcal{C}) = \frac{(\#\text{ honest blocks in } \mathcal{C})}{(\#\text{ total blocks in } \mathcal{C})}.$$

We now show that, under the assumption of zero-network delay ,

$$\text{CQ}(\mathcal{C}) \geq \frac{\lambda_h - \lambda_a}{\lambda_h},$$

so in particular $\text{CQ}(\mathcal{C}) > 0$ whenever $\lambda_h > \lambda_a$, which suffices for liveness.

(i) Let g denote the *chain growth rate* ($\text{CG}(\mathcal{C})$), i.e., the long-term average rate at which blocks appear on the agreed-upon longest chain. Since the adversary controls at most λ_a blocks per unit time, the fraction of adversarial blocks on $\mathcal{C}$ is at most $\frac{\lambda_a}{g}$. Hence,

$$\text{CQ}(\mathcal{C}) = 1 - \frac{(\#\text{ adversary block rate on } \mathcal{C})}{g} \geq \frac{g - \lambda_a}{g}.$$

Since this is true $\forall g \geq 0$,

$$\text{CQ}(\mathcal{C}) \geq \min_{g>0} \frac{g - \lambda_a}{g}$$

(ii) In the zero-delay model, honest miners always build on the current longest chain, so *every* honest-mined block extends $\mathcal{C}$. Therefore, the *chain growth rate* ($\text{CG}(\mathcal{C})$) satisfies $g \geq \lambda_h$.

(iii) We can tighten the bound in (i) by adding the constraint bon g in (ii):

$$\text{CQ}(\mathcal{C}) \geq \min_{g \geq \lambda_h} \frac{g - \lambda_a}{g} \geq 1 - \frac{\lambda_a}{\lambda_h} = \frac{\lambda_h - \lambda_a}{\lambda_h}. \tag{1}$$

Thus, as long as $\lambda_h > \lambda_a$, we have $\text{CQ}(\mathcal{C}) > 0$, and honest blocks continue to appear (and be confirmed) at a positive rate—establishing the *liveness* of the protocol.

We can also rewrite 1 as (for $\beta \leq \frac{1}{2}$):

$$\text{CQ}(\mathcal{C}) \geq \frac{\lambda_h - \lambda_a}{\lambda_h} = \frac{1 - 2\beta}{1 - \beta} \tag{2}$$

(since $\beta = \frac{\lambda_a}{\lambda_a + \lambda_h}$)

Figure 4 illustrates the variation of the computed lower bound with β (has a concave nature).

Continuing from here, the next lecture will involve a discussion about selfish mining.

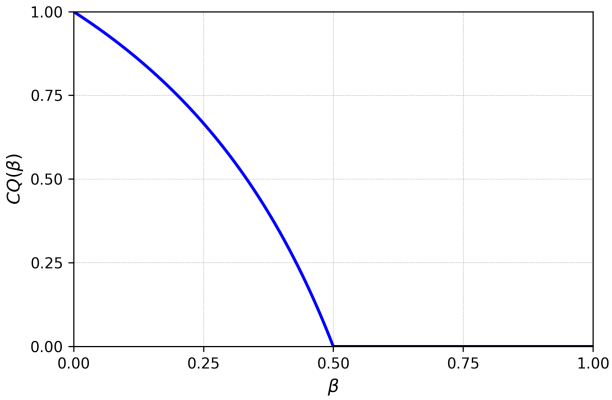


Figure 4: Chain Quality (lower bound) vs. β

References

[1] J. Garay, A. Kiayias, and N. Leonardos. The bitcoin backbone protocol: Analysis and applications. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9057 of *Lecture Notes in Computer Science*, pages 281–310. Springer, 2015.

[2] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. May 2009.

[3] P. Viswanath. Principles of blockchains. `https://pramodv.web.illinois.edu/teaching/ece598pv-spring2021/`, 2021. Course at University of Illinois at Urbana-Champaign, ECE 598PV, Spring 2021.

Appendix: Private Attack Analysis

In our analysis, we assume zero network delay ($\Delta = 0$). As we already saw, honest and adversarial blocks arrive as independent Poisson processes with rates

$$\lambda_h = (1-\beta)\lambda, \qquad \lambda_a = \beta\lambda, \qquad 0 < \beta < \tfrac{1}{2}.$$

Setup: Let $T_i^h \sim \text{Exp}(\lambda_h),\ T_i^a \sim \text{Exp}(\lambda_a),\ i = 0, 1, \ldots, k$, be the inter-arrival times of the *next* $k+1$ blocks after the victim block B' on the honest and private forks, respectively. (The choice $i = 0$ indexes the first block *after* B'.)

The private attack succeeds **iff** the adversary finishes $k{+}1$ blocks *no later* than the honest miners, the below event represents this scenario:

$$\mathcal{E} \;=\; \Big\{ \sum_{i=0}^{k} T_i^h \;>\; \sum_{i=0}^{k} T_i^a \Big\}. \tag{3}$$

By the law-of-large numbers (LLN),

$$\frac{1}{k+1}\sum_{i=0}^{k}(T_i^h - T_i^a) \longrightarrow \frac{1}{\lambda_h} - \frac{1}{\lambda_a} = \frac{1-2\beta}{\beta(1-\beta)\lambda},\ k \to \infty.$$

Hence $\mathcal{E}$ occurs with non-vanishing probability *only* when $\beta > \frac{1}{2}$. For $\beta < \frac{1}{2}$ we quantify the exponentially small tail.

Chernoff/MGF bound

For any $s > 0$,

$$\begin{aligned} Pr[\mathcal{E}] &= Pr\Big[\sum_{i=0}^{k}(T_i^h - T_i^a) > 0\Big] \\ &= Pr\Big[e^{\,s\sum_{i=0}^{k}(T_i^h - T_i^a)} > 1\Big] \\ &\le \mathbf{E}\Big[e^{\,s\sum_{i=0}^{k}(T_i^h - T_i^a)}\Big] \quad \text{(Markov inequality)} \qquad (4) \\ &= \prod_{i=0}^{k}\mathbf{E}\big[e^{\,s(T_i^h - T_i^a)}\big] \quad \text{(independence)} \\ &= \Big(\underbrace{\mathbf{E}[e^{\,sT_0^h}]}_{=\frac{\lambda_h}{\lambda_h - s}} \cdot \underbrace{\mathbf{E}[e^{-sT_0^a}]}_{=\frac{\lambda_a}{\lambda_a + s}}\Big)^{k+1} \\ &= \Big(\frac{\lambda_a}{\lambda_a + s}\cdot\frac{\lambda_h}{\lambda_h - s}\Big)^{k+1}. \qquad (5) \end{aligned}$$

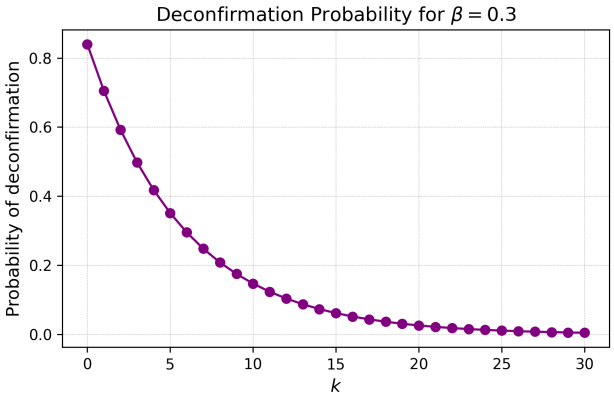
because of the factorization of the moment generating function (MGF) of independent random variables. Now, to optimize the exponent, we choose

$$s^\star \;=\; \frac{(1-2\beta)\lambda}{2} > 0,$$

which lies in $(0, \lambda_h)$ for $\beta < \frac{1}{2}$. Substituting $s = s^\star$ into 5 yields

$$Pr[\mathcal{E}] \le e^{(-c(k+1))}, \text{ where } c = 2\log\Big(\frac{\lambda}{2\sqrt{\beta(1-\beta)\,\lambda}}\Big) \;>\; 0.$$

The below figure illustrates this decay for $\beta = 0.3$ from $k = 0$ to $k = 30$.



Hence the de-confirmation probability decays *exponentially* in the confirmation depth k whenever the adversary controls less than half the hash power. For small β, one may prefer the simpler bound $Pr[\mathcal{E}] \leq (\frac{\beta}{1-\beta})^k$ used in the main text since both expressions are $\Theta\big((\beta/(1-\beta))^k\big)$ as $k \to \infty$.